Preprint
UCRL-JC-136574

# Generalized B-Spline Subdivision-Surface Wavelets and Lossless Compression

*M. Bertram, M. A. Duchaineau, B. Hamann, K. I. Joy*

This article was submitted to
Institute of Electrical and Electronics Engineers Incorporated 2000
Symposium on Visualization
Amsterdam, The Netherlands
May 29-31, 2000

*U.S. Department of Energy*

Lawrence
Livermore
National
Laboratory

**November 24, 1999**

This work was performed under the auspices of the United States Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

# Generalized B-Spline Subdivision-Surface Wavelets and Lossless Compression

Martin Bertram*† Mark A. Duchaineau†
Bernd Hamann* Kenneth I. Joy*

November 24, 1999

**Keywords:** arbitrary topology, compression, computer-aided geometric design, contours, geometry compression, isosurfaces, lossless compression, multiresolution methods, subdivision surfaces, symmetric wavelets, wavelet lifting.

## Abstract

We present a new construction of wavelets on arbitrary two-manifold topology for geometry compression. The constructed wavelets generalize symmetric tensor product wavelets with associated B-spline scaling functions to irregular polygonal base mesh domains. The wavelets and scaling functions are tensor products almost everywhere, except in the neighborhoods of some extraordinary points (points of valence unequal four) in the base mesh

*Center for Image Processing and Integrated Computing (CIPIC), Department of Computer Science, University of California, Davis, CA 95616-8562, USA

†Center for Applied Scientific Computing (CASC), Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551, USA

that defines the topology. The compression of arbitrary polygonal meshes representing isosurfaces of scalar-valued trivariate functions is a primary application. The main contribution of this paper is the generalization of lifted symmetric tensor product B-spline wavelets to two-manifold geometries. Surfaces composed of B-spline patches can easily be converted to this scheme. We present a lossless compression method for geometries with or without associated functions like color, texture, or normals. The new wavelet transform is highly efficient and can represent surfaces at any level of resolution with high degrees of continuity, except at a finite number of extraordinary points in the base mesh. In the neighborhoods of these points detail can be added to the surface to approximate any degree of continuity.

# 1 Introduction

Representing two-manifold geometries like isosurfaces, shock waves or material boundaries in three-dimensional scalar or vector fields is an important problem in scientific visualization. Surface representations need to provide efficient access to local geometry satisfying any bounds on error or complexity that are defined by a query. For storage and transmission purposes, compressed representations are required to reduce storage space and transmission time.

Wavelet representations [3, 4, 15, 24] satisfy many of these requirements. The discrete wavelet transform (DWT) [18] is used in lossy and lossless compression schemes for image and terrain data to obtain high compression rates. Details defined by high frequencies are successively separated from the remaining spectrum of lower frequencies by representing an image as a two-variate function in a wavelet basis, see Figure 1. Compression is obtained by arithmetic encoding [20] of the wavelet coefficients that are sparse or have low absolute values.

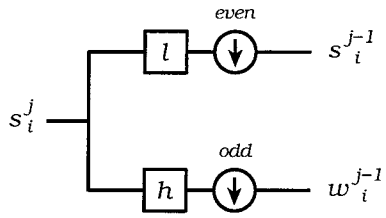


Figure 1: Wiring diagram for DWT. The scaling function coefficients $s_i^j$ at level $j$ are obtained by low pass filtering and downsampling. The wavelet coefficients $w_i^j$ are obtained by separating the highest frequency band and downsampling.

For compression purposes the DWT is superior to the discrete Fourier and cosine transforms, since it has efficiency $O(n)$, where n is the number of pixels or samples, and it can localize features in both signal and frequency space. There exists a variety of different band pass and low pass filters with corresponding basis functions (wavelets and scaling functions) that can be used to implement a DWT. In most compression schemes used for image, terrain or volume data, tensor product wavelets are used, i.e., a one-dimensional wavelet transform is applied successively in the different dimensions of a dataset. For more details about the wavelet transform and the construction of wavelets with different properties, such as orthogonality, vanishing moments, and compact support, we refer to [3, 4, 7, 12, 13, 15, 18, 19, 24, 25, 27].

Wavelets on arbitrary topology have been initially explored by Lounsbery [16] and by Schröder and Sweldens [21]. Though Lounsbery pointed out that wavelets can be constructed for any type of subdivision surface, most approaches are based on triangular subdivision surfaces like the *butterfly scheme* [8] and *Loop's scheme* [17]. In the work presented here, we describe a construction of wavelets based on subdivision surfaces similar to *Catmull-Clark surfaces* [2, 6, 22, 23] that generalize tensor product B-splines.

The paper is structured as follows: In Section 2, we define the lifting operations that are used to construct symmetric one-dimensional wavelets asso-

ciated with B-spline scaling functions. In Section 3, these lifting operations are generalized to polygonal meshes so that the wavelets from Section 2 can be constructed for these polygonal meshes. In the case of a rectilinear mesh, the constructed basis functions are tensor products. We present a lossless geometry compression algorithm and provide numerical results in Section 4. Directions for future research conclude the work in Section 5.

# 2 Symmetric Lifted Wavelets

*Wavelet lifting* was introduced by Sweldens [26], and it is often used for wavelet construction, see [1, 14]. The idea is to decompose a single filtering step of the DWT into small local filtering operations. Lifting can increase the efficiency of the transform, make it feasible to use integer arithmetic for lossless compression, and simplify the construction of wavelets. In this section, we provide a class of lifted wavelets that can be generalized to polygonal mesh domains. We start with reviewing some basics about the DWT.

## 2.1 Discrete Wavelet Transform

The DWT is a basis transform between certain spaces spanned by dilated and translated versions of a wavelet $\psi$ and a scaling function $\phi$:

$$\phi_i^j(x) \;=\; \phi(2^j x - i) \quad \text{and} \quad \psi_i^j(x) \;=\; \psi(2^j x - i). \tag{2.1}$$

A function $f$ that is initially represented in a basis of scaling functions at a level $j_\varepsilon > 0$ as

$$f(x) \;=\; \sum_{i\in\mathbb{Z}} s_i^{j_\varepsilon}\; \phi_i^{j_\varepsilon}(x), \tag{2.2}$$

is decomposed into a basis of wavelets and scaling functions at level $j_\varepsilon - 1$,

$$f(x) \;=\; \sum_{i\in\mathbb{Z}} w_i^{j_\varepsilon-1}\; \psi_i^{j_\varepsilon-1}(x) \;+\; \sum_{i\in\mathbb{Z}} s_i^{j_\varepsilon-1}\; \phi_i^{j_\varepsilon-1}(x). \tag{2.3}$$

This transform is called *decomposition*. Decomposition steps are successively applied to the representation by scaling functions at the next coarser levels until a base level $j = 0$ is reached. The function $f$ is finally represented as

$$f(x) = \sum_{j=0}^{j_\varepsilon - 1} \sum_{i \in \mathbb{Z}} w_i^j \, \psi_i^j(x) + \sum_{i \in \mathbb{Z}} s_i^0 \, \phi_i^0(x). \tag{2.4}$$

Figure 2 is an illustration of this basis transform.

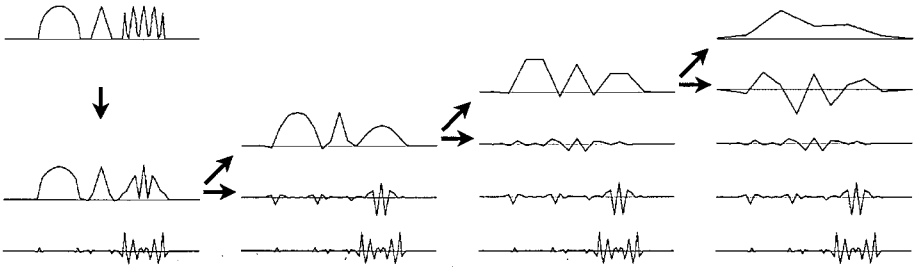

Figure 2: Decomposition steps of a DWT. The function in the upper left is transformed by successive decomposition steps, using a linear B-spline wavelet.

A decomposition step is implemented by a discrete filtering with sequences $h$ and $l$ that transform scaling function coefficients $s_i^j$ at level $j$ into scaling function coefficients $s_i^{j-1}$ and wavelet coefficients $w_i^{j-1}$ at level $j-1$. The decomposition rules are defined as

$$s_i^{j-1} = \sum_{k \in \mathbb{Z}} l_{k-2i} \, s_k^j \quad \text{and} \tag{2.5}$$

$$w_i^{j-1} = \sum_{k \in \mathbb{Z}} h_{k-2i} \, s_k^j. \tag{2.6}$$

For the inverse DWT each individual decomposition step is inverted by a *reconstruction* step using filters $\tilde{h}$ and $\tilde{l}$,

$$s_i^j = \sum_{k \in \mathbb{Z}} \left( \tilde{l}_{i-2k} \, s_k^{j-1} + \tilde{h}_{i-2k} \, w_k^{j-1} \right). \tag{2.7}$$

Applying reconstruction steps in reverse order of corresponding decomposition steps reproduces the initial representation defined by scaling functions (2.2).

## 2.2 Lifting Approach

Rather then computing the coefficients defined in equations (2.5–2.7) directly, we subdivide the summation steps into local *lifting operations* that can be computed more efficiently. We describe the lifting scheme illustrated in Figure 3 using algorithmic notation. A decomposition step for the DWT is defined by re-labeling coefficients

$$s_i^{j-1} \leftarrow s_{2i}^j \quad \text{and} \quad w_i^{j-1} \leftarrow s_{2i+1}^j \tag{2.8}$$

followed by a sequence of alternating *s-lift* and *w-lift* operations. These operations modify one coefficient at a time, depending on its own and its two neighbors' values. The s-lift and w-lift operations are defined as

**s-lift**$(a, b)$**:**

$$s_i^{j-1} \leftarrow a w_{i-1}^{j-1} + b s_i^{j-1} + a w_i^{j-1} \quad \forall i \tag{2.9}$$

**w-lift**$(a, b)$**:**

$$w_i^{j-1} \leftarrow a s_i^{j-1} + b w_i^{j-1} + a s_{i+1}^{j-1} \quad \forall i \tag{2.10}$$

Since we use the same lifting operations to construct both filters $h$ and $l$ at once, we restrict the class of wavelets that can be defined by these operations. However, we gain efficiency by using this specific lifting scheme and, as we show in Section 3, the s-lift and w-lift operations generalize nicely to polygonal meshes. Another advantage is that each lifting step is inverted by replacing $a$ and $b$ by $-\frac{a}{b}$ and $\frac{1}{b}$, respectively, see Figure 4. The only constraint for the existence of the inverse DWT is that the parameter $b$ must be non-zero for all lifting operations.

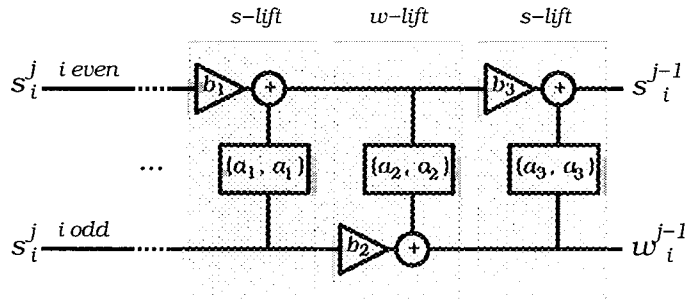


Figure 3: Wiring diagram for lifted DWT. Lifting operations that modify coefficients with even indices are called *s-lift* and those that modify coefficients with odd indices are called *w-lift* operations.

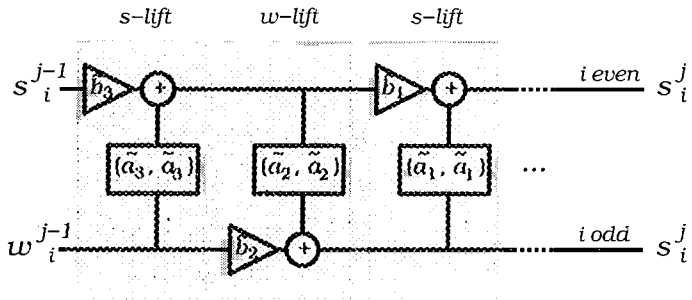


Figure 4: Wiring diagram for inverse DWT. The same lifting operations as for DWT occur in reverse order with changed parameters $\tilde{a}_i = -\frac{a_i}{b_i}$ and $\tilde{b}_i = \frac{1}{b_i}$.

## 2.3 B-Spline Wavelet Construction

We choose B-splines as scaling functions due to their wide applicability and use in computer-aided geometric design (CAGD) and approximation theory. From the two-scale relation for B-splines [7], the reconstruction filter $\tilde{l}$ is given by

$$\phi(x) = \sum_{i \in \mathbb{Z}} \tilde{l}_i \, \phi(2x - i). \tag{2.11}$$

The non-zero values for $\tilde{l}_i$ can be obtained from Pascal's triangle by dividing the entries in the $(n+2)$th row by $2^n$, where $n$ is the polynomial degree, see

Table 1. Since the number of non-zero coefficients must be odd for our lifting scheme, we can only construct the DWT for B-spline scaling functions with odd polynomial degrees.

| degree | $\tilde{l}_0$ | $\tilde{l}_{\pm 1}$ | $\tilde{l}_{\pm 2}$ | $\tilde{l}_{\pm 3}$ |
|---|---|---|---|---|
| linear | 1 | $\frac{1}{2}$ | | |
| cubic | $\frac{3}{4}$ | $\frac{1}{2}$ | $\frac{1}{8}$ | |
| quintic | $\frac{5}{8}$ | $\frac{15}{32}$ | $\frac{3}{16}$ | $\frac{1}{32}$ |

Table 1: Coefficients for two-scale relation of B-splines.

Analogously, a wavelet can be defined as a linear combination of scaling functions

$$\psi(x) \;=\; \sum_{i\in\mathbb{Z}} \tilde{h}_i \; \phi(2x-i). \tag{2.12}$$

To ensure that the constructed wavelet has zero *direct current* (average obtained from integration)[1], filter $\tilde{h}$ must satisfy the condition

$$\sum_{i\in\mathbb{Z}} \tilde{h}_i \;=\; 0. \tag{2.13}$$

### 2.3.1 Linear B-Spline Wavelets

The construction of $\tilde{h}$ is constrained by our specific lifting approach. Figure 5 depicts the lifting scheme for a linear B-spline wavelet, and Figure 6 shows the constraints that result from the lifting scheme. These constraints for $\tilde{l}$ are

$$\tilde{l}_0 \;=\; \tilde{b}_2 \quad \text{and} \quad \tilde{l}_1 \;=\; \tilde{b}_2\tilde{a}_1. \tag{2.14}$$

[1]This condition is necessary for the continuous wavelet transform [3], but not for the lifted DWT. However, it is useful to reduce the magnitude of wavelet coefficients that represent variations of a function rather then averages.

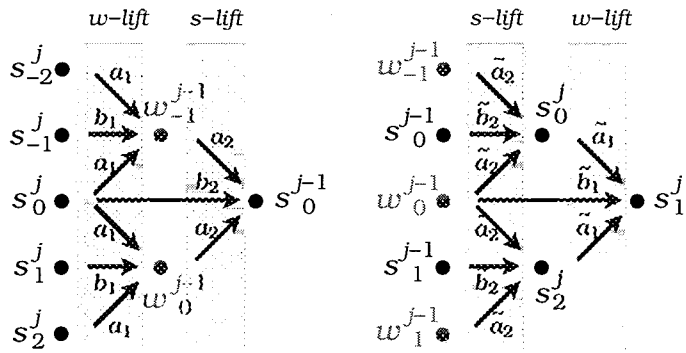


Figure 5: Lifting scheme for linear B-spline wavelet. Decomposition (left) and reconstruction (right) are composed of one s-lift and one w-lift operation.

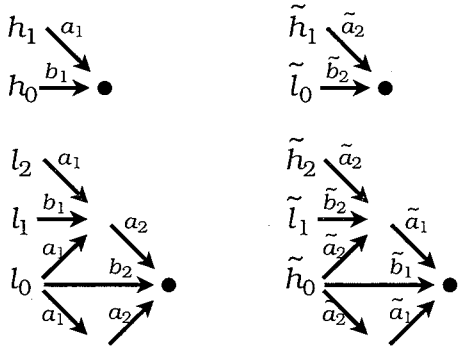

Figure 6: Constraints for construction of filters $h$, $l$, $\tilde{h}$, and $\tilde{l}$ for linear B-spline scaling functions.

Using the values from Table 1, we obtain

$$a_1 = \tfrac{1}{2} \quad \text{and} \quad b_2 = 1. \tag{2.15}$$

The constraints for $\tilde{h}$ are given by

$$\tilde{h}_0 = 2\tilde{a}_2\tilde{a}_1 + \tilde{b}_1 = \tilde{a}_2 + \tilde{b}_1, \tag{2.16}$$

$$\tilde{h}_1 = \tilde{a}_2, \quad \text{and} \tag{2.17}$$

$$\tilde{h}_2 = \tilde{a}_2\tilde{a}_1 = \tfrac{1}{2}\tilde{a}_2. \tag{2.18}$$

We note that $\tilde{l}$ and $\tilde{h}$ are symmetric, and the coefficients with negative indices do not produce additional constraints. Hence, (2.13) becomes

$$\tilde{h}_0 + 2\tilde{h}_1 + 2\tilde{h}_2 = 4\tilde{a}_2 + \tilde{b}_1 = 0. \tag{2.19}$$

Since $\tilde{a}_2$ and $\tilde{b}_1$ are proportional and either one of them appears on the right-hand sides of equations (2.16–2.18), they do not modify $\tilde{h}$, except for scaling it. Thus, we can choose $\tilde{b}_1 = 1$ and obtain the lifting parameters shown in Table 2. The remaining filters $h$ and $l$ can be derived from these lifting parameters, which are summarized in Table 3.

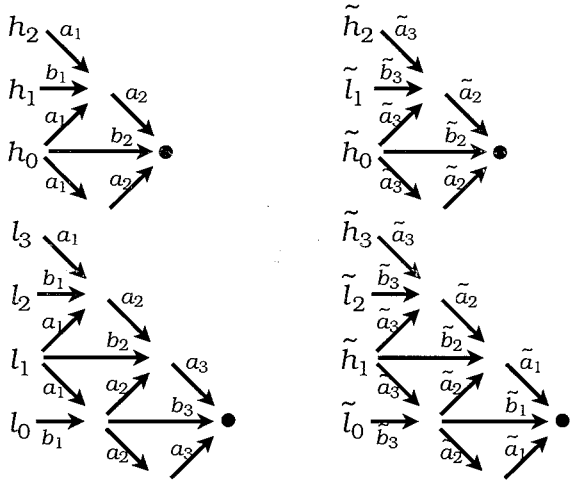


Figure 7: Constraints for lifting scheme for cubic B-spline scaling functions.

### 2.3.2 Cubic B-Spline Wavelets

In analogy to linear B-spline wavelets, one can construct wavelets for cubic scaling functions using one additional lifting step. The constraints for the lifting parameters are shown in Figure 7. The constraints for $\tilde{l}$ are equivalent to

$$\tilde{a}_1 = \tfrac{1}{4}, \quad \tilde{a}_2 = \tilde{b}_1, \quad \text{and} \quad \tilde{b}_1\tilde{b}_3 = \tfrac{1}{2}. \tag{2.20}$$

The equations for $\tilde{h}$, after eliminating $\tilde{a}_1$ and $\tilde{a}_2$, are given by

$$\tilde{h}_0 = \tilde{b}_2 + 2\tilde{a}_3\tilde{b}_1, \tag{2.21}$$

$$\tilde{h}_1 = \tfrac{1}{4}\tilde{b}_2 + \tfrac{7}{4}\tilde{a}_3\tilde{b}_1 \tag{2.22}$$

$$\tilde{h}_2 = \tilde{a}_3\tilde{b}_1, \quad \text{and} \tag{2.23}$$

$$\tilde{h}_3 = \tfrac{1}{4}\tilde{a}_3\tilde{b}_1. \tag{2.24}$$

Using equation (2.13) we obtain

$$3\tilde{b}_2 = -16\tilde{a}_3\tilde{b}_1. \tag{2.25}$$

Again, we observe that the remaining lifting parameters do not modify $\tilde{h}$, except for scaling, since $\tilde{b}_2$ and $\tilde{a}_3$ (as well as $\tilde{b}_2$ and $\tilde{b}_1$) are proportional. Hence, we can choose $\tilde{b}_1 = \tilde{b}_2 = 1$ and determine the remaining parameters from the above equations. The resulting values are listed in Tables 2 and 3.

### 2.3.3 Quintic B-Spline Wavelets

For the quintic case, we only summarize the solution. From the constraints for $\tilde{l}$ we obtain the equations

$$\tilde{a}_1 = \tfrac{1}{6}, \tag{2.26}$$

$$\tilde{a}_2 = \tfrac{9}{16}\tilde{b}_1, \tag{2.27}$$

$$\tilde{b}_4\tilde{b}_2 = \tfrac{1}{4}, \quad \text{and} \tag{2.28}$$

$$\tilde{a}_3\tilde{b}_1 = \tfrac{4}{3}\tilde{b}_2. \tag{2.29}$$

The constraints for $\tilde{h}$ can be written in terms of $\tilde{b}_3\tilde{b}_1$ and $\tilde{a}_4\tilde{b}_2$. Inserting them into equation (2.13) results in

$$5\tilde{b}_3\tilde{b}_1 = -32\tilde{a}_4\tilde{b}_2. \tag{2.30}$$

We can choose $\tilde{b}_3$, $\tilde{b}_2$, and $\tilde{b}_1$ to be one, since they do not modify $\tilde{h}$. The remaining lifting parameters are then uniquely determined.

The resulting lifting parameters and filters for our constructions are summarized in Tables 2 and 3. The constructed wavelets are plotted in Figure 8. A wider class of wavelets is available by using scaling functions different from B-splines. In particular, it is feasible to construct a wavelet first and obtain a proper scaling function from the constraints given by the lifting scheme.

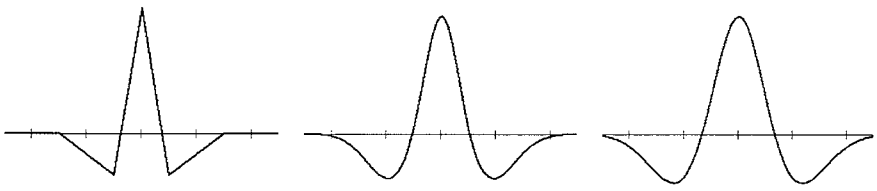

Figure 8: Linear, cubic, and quintic B-spline wavelets.

| degree | $a_1$ | $b_1$ | $a_2$ | $b_2$ | $a_3$ | $b_3$ | $a_4$ | $b_4$ |
|---|---|---|---|---|---|---|---|---|
| linear | $-\frac{1}{2}$ | 1 | $\frac{1}{4}$ | 1 | | | | |
| cubic | $-\frac{1}{4}$ | 1 | $-1$ | 1 | $\frac{3}{8}$ | 2 | | |
| quintic | $-\frac{1}{6}$ | 1 | $-\frac{9}{16}$ | 1 | $-\frac{4}{3}$ | 1 | $\frac{5}{8}$ | 4 |

| degree | $\tilde{a}_1$ | $\tilde{b}_1$ | $\tilde{a}_2$ | $\tilde{b}_2$ | $\tilde{a}_3$ | $\tilde{b}_3$ | $\tilde{a}_4$ | $\tilde{b}_4$ |
|---|---|---|---|---|---|---|---|---|
| linear | $\frac{1}{2}$ | 1 | $-\frac{1}{4}$ | 1 | | | | |
| cubic | $\frac{1}{4}$ | 1 | 1 | 1 | $-\frac{3}{16}$ | $\frac{1}{2}$ | | |
| quintic | $\frac{1}{6}$ | 1 | $\frac{9}{16}$ | 1 | $\frac{4}{3}$ | 1 | $-\frac{5}{32}$ | $\frac{1}{4}$ |

Table 2: Lifting parameters for DWT and inverse DWT.

| degree | $h_0$ | $h_{\pm 1}$ | $h_{\pm 2}$ | $h_{\pm 3}$ | $l_0$ | $l_{\pm 1}$ | $l_{\pm 2}$ | $l_{\pm 3}$ | $l_{\pm 4}$ |
|---|---|---|---|---|---|---|---|---|---|
| linear | 1 | $-\frac{1}{2}$ | | | $\frac{3}{4}$ | $\frac{1}{4}$ | $-\frac{1}{8}$ | | |
| cubic | $\frac{3}{2}$ | $-1$ | $\frac{1}{4}$ | | $\frac{5}{4}$ | $\frac{5}{32}$ | $-\frac{3}{8}$ | $\frac{3}{32}$ | |
| quintic | $\frac{5}{2}$ | $-\frac{15}{8}$ | $\frac{3}{4}$ | $-\frac{1}{8}$ | $\frac{231}{96}$ | $-\frac{7}{32}$ | $-\frac{21}{24}$ | $\frac{15}{32}$ | $-\frac{5}{64}$ |

| degree | $\tilde{h}_0$ | $\tilde{h}_{\pm 1}$ | $\tilde{h}_{\pm 2}$ | $\tilde{h}_{\pm 3}$ | $\tilde{h}_{\pm 4}$ |
|---|---|---|---|---|---|
| linear | $\frac{3}{4}$ | $-\frac{1}{4}$ | $-\frac{1}{8}$ | | |
| cubic | $\frac{5}{8}$ | $-\frac{5}{64}$ | $-\frac{3}{16}$ | $-\frac{3}{64}$ | |
| quintic | $\frac{77}{128}$ | $\frac{7}{128}$ | $-\frac{7}{32}$ | $-\frac{15}{128}$ | $-\frac{5}{256}$ |

Table 3: Filters for DWT and inverse DWT resulting from lifting parameters.

# 3 Generalization to Arbitrary Polygonal Meshes

The preceding section describes basic lifting operations and constructions of one-dimensional wavelets based on these operations. In this section, we generalize the w-lift and s-lift operations to polygonal meshes so that they reproduce tensor product wavelets and scaling functions for rectilinear base meshes.

## 3.1 Index-free Notation for Subdivision Rules

Subdivision surfaces are limit-surfaces that result from recursive subdivision of polygonal base meshes. A subdivision step refines a *submesh* to a finer mesh, called *supermesh*. Positions for all vertices the supermesh are computed from the positions of the vertices in the submesh, based on certain subdivision rules. Usually, a new vertex position depends linearly on a local set of vertices. Most subdivision schemes converge rapidly to a continuous limit-surface so that a mesh obtained from just a few subdivision steps is often a good approximation for surface rendering. In many cases, parametrization and exact evaluation of limit-surface points is possible, see [23].

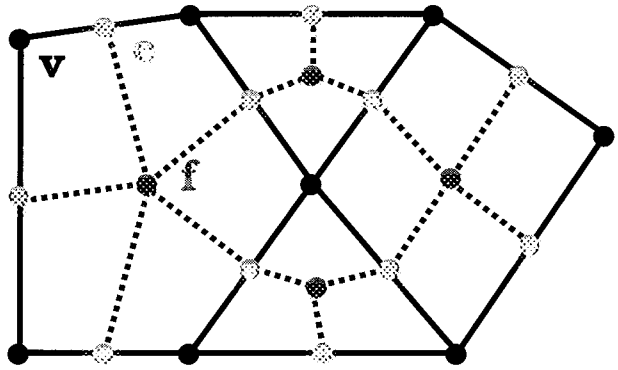


Figure 9: Topology for Catmull-Clark subdivision. The submesh is drawn in bold lines and the vertices of the supermesh are denoted as $\mathbf{f}$, $\mathbf{e}$, and $\mathbf{v}$.

For our approach, we use a subdivision mesh topology as defined by Catmull-Clark subdivision [2]. vertices in the supermesh correspond to a face (polygon), an edge, or a vertex in the submesh. We denote these corresponding vertex types as $\mathbf{f}$, $\mathbf{e}$, and $\mathbf{v}$, respectively, see Figure 9.

To describe subdivision rules that determine the new vertex positions, we introduce a notation that is index-free and thus intuitive. We use the averaging operator $\overline{x}_y$, where $x$ and $y$ can represent $\mathbf{f}$, $\mathbf{e}$, or $\mathbf{v}$. This operator returns the arithmetic average of all vertices of type $x$ that are adjacent to $y$ in a supermesh. If there are no direct neighbors of type $x$, e.g., $x$ and $y$ represent the same symbol, then we compute the average for the set of first neighbors with respect of adjacent primitives in the submesh. Examples for the index-free notation are depicted in Figure 10.

A subdivision scheme that generalizes bilinear B-Splines [10] is defined by the subdivision rules

$$\mathbf{f} \leftarrow \overline{\mathbf{v}}_{\mathbf{f}}, \quad \mathbf{e} \leftarrow \overline{\mathbf{v}}_{\mathbf{e}}, \quad \text{and} \quad \mathbf{v} \leftarrow \mathbf{v}. \tag{3.1}$$

This means that each vertex $\mathbf{f}$ is the centroid of its associated face, each vertex $\mathbf{e}$ is the midpoint of its associated edge, and each vertex $\mathbf{v}$ is the same point as in the submesh.

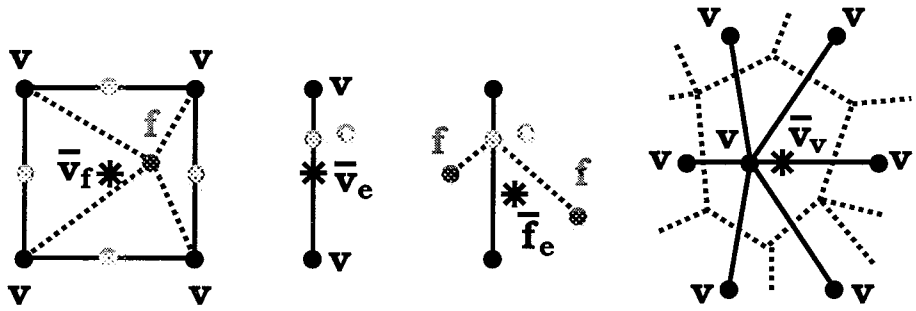


Figure 10: Examples for index-free notation. From left to right: $\overline{\mathbf{v}}_\mathbf{f}$ is the centroiod of a face; $\overline{\mathbf{v}}_\mathbf{e}$ is the midpoint of an edge; $\overline{\mathbf{f}}_\mathbf{e}$ is the midpoint of the line segment connecting the vertices $\mathbf{f}$ belonging to adjacent faces; $\overline{\mathbf{v}}_\mathbf{v}$ is the centroid of all adjacent vertices $\mathbf{v}$ in the submesh (since there are no direct neighbors in the supermesh).

A formulation of Catmull-Clark subdivision in index-free notation is given by the rules

$$\mathbf{f} \leftarrow \overline{\mathbf{v}}_\mathbf{f}, \quad \mathbf{e} \leftarrow \tfrac{1}{2}\left(\overline{\mathbf{v}}_\mathbf{e} + \overline{\mathbf{f}}_\mathbf{e}\right), \quad \text{and} \quad \mathbf{v} \leftarrow \tfrac{1}{n_\mathbf{v}}\left(\overline{\mathbf{f}}_\mathbf{v} + \overline{\mathbf{v}}_\mathbf{v} + (n_\mathbf{v} - 2)\mathbf{v}\right), \tag{3.2}$$

where $n_\mathbf{v}$ is the valence (number of incident edges) of a vertex.

## 3.2 Generalized Lifting Operations

In the case of a tensor product wavelet transform, we apply a decomposition step of the one-dimensional DWT to all rows and then columns of a dataset, see Figure 11. This results in sets of coefficients $s$, $w_1$, $w_2$, $w_3$ for four different types of basis functions, which we denote as

$$\begin{aligned} \phi(x,y) &= \phi(x)\phi(y), \\ \psi_1(x,y) &= \phi(x)\psi(y), \\ \psi_2(x,y) &= \psi(x)\phi(y), \textit{and} \end{aligned}$$

$$\psi_3(x,y) \;=\; \psi(x)\psi(y).$$

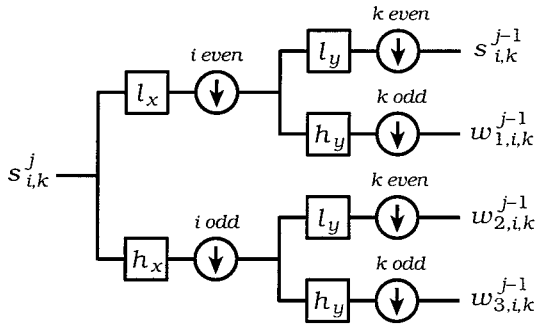


Figure 11: Wiring diagram for two-dimensional DWT. Coefficients are labeled with consecutive indices $i$ in $x$-direction and $k$ in $y$-direction.

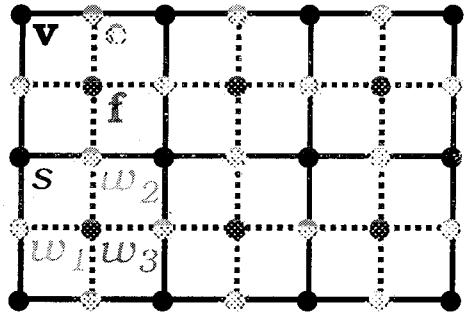


Figure 12: Rectilinear mesh and correspondence between vertices and wavelet coefficients.

A decomposition step can be considered as an operation applied to a rectilinear supermesh that computes vertex positions for an approximating submesh and replaces the remaining supermesh vertices by some difference vectors representing details that do not exist in the submesh. Vertices **v** represent coefficients $s$ for scaling functions, vertices **e** represent wavelet coefficients $w_1$ and $w_2$ (depending on the orientation of edges), and vertices **f** represent wavelet coefficients $w_3$, see Figure 12.

Instead of applying an entire one-dimensional decomposition step first to all rows and subsequently to all columns of a dataset, it is possible to do this

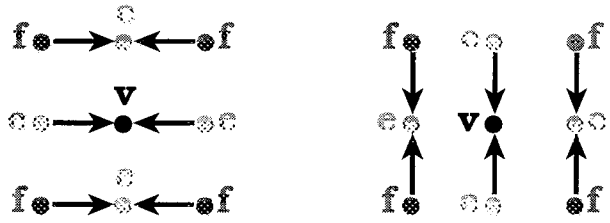


Figure 13: An s-lift operation applied to a rectilinear grid is composed of a vertical (left) and a horizontal (right) one-dimensional s-lift operation.

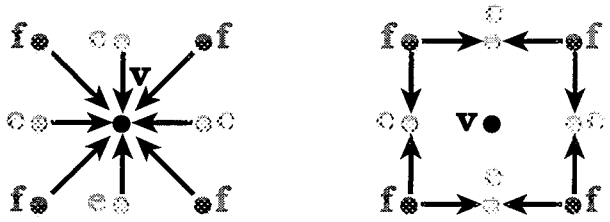


Figure 14: Same s-lift operation as shown in Figure 13 with different order of vertex modifications. First, vertices $\mathbf{v}$ are modified (left) and then vertices $\mathbf{e}$ (right).

individually for each lifting operation. Figure 13 illustrates this approach for an *s*-lift operation. However, the same operation can be computed by modifying the vertices $\mathbf{v}$ and $\mathbf{e}$ separately, see Figure 14. Analogously, a w-lift operation can be computed by modifying the vertices $\mathbf{f}$ and $\mathbf{e}$ separately. Both lifting operations can therefore be defined as follows:

**s-lift**$(a, b)$**:**

$$\mathbf{v} \leftarrow b^2\mathbf{v} + 4a^2\overline{\mathbf{f}}_{\mathbf{v}} + 4ab\overline{\mathbf{e}}_{\mathbf{v}} \quad \text{and} \quad \mathbf{e} \leftarrow b\mathbf{e} + 2a\overline{\mathbf{f}}_{\mathbf{e}} \tag{3.3}$$

**w-lift**$(a, b)$**:**

$$\mathbf{f} \leftarrow b^2\mathbf{f} + 4a^2\overline{\mathbf{v}}_{\mathbf{f}} + 4ab\overline{\mathbf{e}}_{\mathbf{f}} \quad \text{and} \quad \mathbf{e} \leftarrow b\mathbf{e} + 2a\overline{\mathbf{v}}_{\mathbf{e}} \tag{3.4}$$

These lifting operations are defined in a notation suitable for arbitrary polygonal meshes that defining two-manifold surfaces with or without boundaries

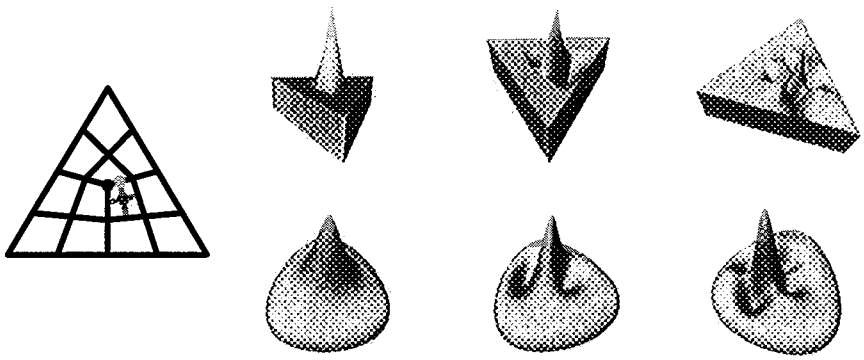

Figure 15: Basis functions around an extraordinary vertex of valence three. Far left: mesh configuration (top face of base mesh with control points corresponding to basis functions). Top row: linear construction; bottom row: cubic construction; From left to right: scaling function, wavelet corresponding to an edge, wavelet corresponding to a face.

(they can even be applied to non-manifold geometries). For each modification step, the overall weight of adjacent vertices that is added to the modified vertex depends only on the lifting parameters $a$ and $b$, but it does not depend on the number of adjacent vertices. Thus, wavelet coefficients that are located closely to extraordinary points may not become larger on average, than wavelet coefficients located in rectilinear areas. We have found that the percentage of coefficients affected by extraordinary points is very small for the finer levels of detail.

We now generalize the wavelets constructed in Section 2 to polygonal meshes, since they are completely defined by s-lift and w-lift operations. These can be inverted in the same way as in the one-dimensional case so that an efficient algorithm for the inverse DWT can be developed. Examples for the constructed basis functions are shown in Figures 15–17.

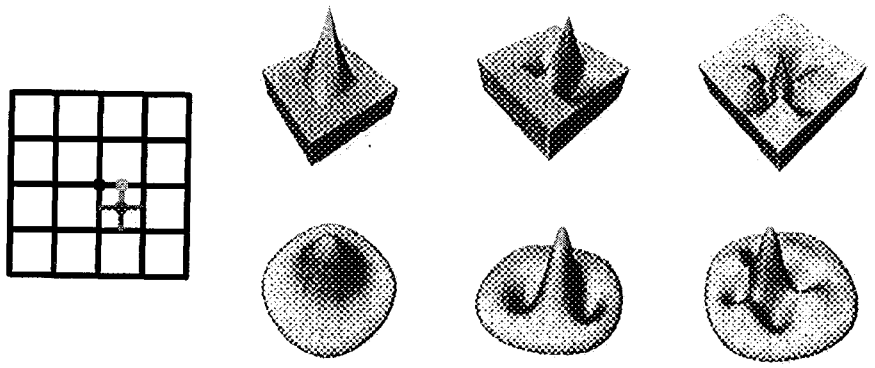

Figure 16: Basis functions for rectilinear case.

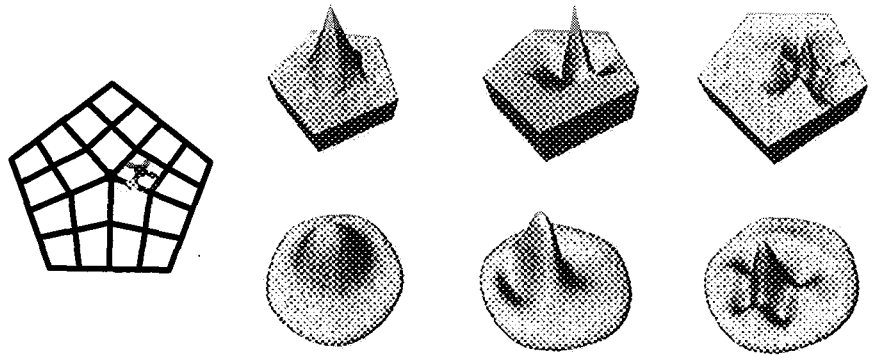

Figure 17: Basis functions around an extraordinary vertex of valence five.

# 4 Lossless Geometry Compression

In this section, we construct a wavelet transform that maps integers to integers and can be inverted without loss. The construction is based on cubic B-spline wavelets. Similar approaches are feasible for linear and the quintic wavelets.

We assume (i) that a surface can be approximated by a mesh topology obtained from a base mesh by a number of subdivisions and (ii) that vertex coordinates are represented as integers. For a surface representation defined by floating-point numbers, a prescribed tolerance $\varepsilon$ can be maintained by scaling all coordinates by $\frac{1}{2\varepsilon}$ and rounding to the closest integer. For algorithms that reconstruct a base mesh and provide a network of approximating B-spline patches starting with a set of discrete surface samples we refer to Eck and Hoppe [9] and Guo [11].

The condition for the existence of an invertable integer-to-integer DWT is: all lifting parameters $b_i$ must be integers. This is the case for the wavelets constructed in Section 2. Thus, the lifting operations defined by equations (3.3) and (3.4) can be approximated as follows:

**integer s-lift**$(a, b)$**:**

$$\mathbf{v} \leftarrow b^2\mathbf{v} + \lfloor \tfrac{1}{2} + 4a^2\overline{\mathbf{f}}_{\mathbf{v}} + 4ab\overline{\mathbf{e}}_{\mathbf{v}} \rfloor \quad \text{and} \quad \mathbf{e} \leftarrow b\mathbf{e} + \lfloor \tfrac{1}{2} + 2a\overline{\mathbf{f}}_{\mathbf{e}} \rfloor \quad (4.1)$$

**integer w-lift**$(a, b)$**:**

$$\mathbf{f} \leftarrow b^2\mathbf{f} + \lfloor \tfrac{1}{2} + 4a^2\overline{\mathbf{v}}_{\mathbf{f}} + 4a\overline{\mathbf{e}}_{\mathbf{f}} \rfloor \quad \text{and} \quad \mathbf{e} \leftarrow b\mathbf{e} + \lfloor \tfrac{1}{2} + 2a\overline{\mathbf{v}}_{\mathbf{e}} \rfloor \quad (4.2)$$

These integer lifting operations can be inverted without loss of precision by subtracting the term that has been added to a vertex and by reversing the order of the individual vertex modifications:

**inverse integer s-lift**$(a, b)$**:**

$$\mathbf{e} \leftarrow \tfrac{1}{b}\left(\mathbf{e} - \lfloor \tfrac{1}{2} + 2a\overline{\mathbf{f}}_{\mathbf{e}} \rfloor\right) \quad \text{and} \quad \mathbf{v} \leftarrow \tfrac{1}{b^2}\left(\mathbf{v} - \lfloor \tfrac{1}{2} + 4a^2\overline{\mathbf{f}}_{\mathbf{v}} + 4a\overline{\mathbf{e}}_{\mathbf{v}} \rfloor\right) \tag{4.3}$$

**inverse integer w-lift**$(a, b)$**:**

$$\mathbf{e} \leftarrow \tfrac{1}{b}\left(\mathbf{e} - \lfloor \tfrac{1}{2} + 2a\overline{\mathbf{v}}_{\mathbf{e}} \rfloor\right) \quad \text{and} \quad \mathbf{f} \leftarrow \tfrac{1}{b^2}\left(\mathbf{f} - \lfloor \tfrac{1}{2} + 4a^2\overline{\mathbf{v}}_{\mathbf{f}} + 4a\overline{\mathbf{e}}_{\mathbf{f}} \rfloor\right) \tag{4.4}$$

Using these integer lifting operations, we now construct the integer DWT for the cubic B-spline wavelet explicitly. The DWT can be defined by the sequence

$$\mathrm{s-lift}\left(-\tfrac{1}{4}, 1\right), \quad \mathrm{w-lift}\left(-1, 1\right), \quad \text{and} \quad \mathrm{s-lift}\left(-\tfrac{3}{8}, 2\right).$$

To keep wavelet coefficients as small as possible, we divide the lifting parameters of the last s-lift operation by two, which corresponds to loosing one bit of precision at each decomposition step. One looses two bits of precision for the two-dimensional transform. Inserting equations (4.1) and (4.2) leads to the decomposition formula that is composed of six vertex modifications, given by

$$\begin{aligned}
\mathbf{v} &\leftarrow \mathbf{v} + \lfloor \tfrac{1}{2} + \tfrac{1}{4}\overline{\mathbf{f}}_{\mathbf{v}} - \overline{\mathbf{e}}_{\mathbf{v}} \rfloor \\
\mathbf{e} &\leftarrow \mathbf{e} + \lfloor \tfrac{1}{2} - \tfrac{1}{2}\overline{\mathbf{f}}_{\mathbf{e}} \rfloor \\
\mathbf{f} &\leftarrow \mathbf{f} + \lfloor \tfrac{1}{2} + 4\overline{\mathbf{v}}_{\mathbf{f}} - 4\overline{\mathbf{e}}_{\mathbf{f}} \rfloor \\
\mathbf{e} &\leftarrow \mathbf{e} + \lfloor \tfrac{1}{2} - 2\overline{\mathbf{v}}_{\mathbf{e}} \rfloor \\
\mathbf{v} &\leftarrow \mathbf{v} + \lfloor \tfrac{1}{2} + \tfrac{9}{64}\overline{\mathbf{f}}_{\mathbf{v}} + \tfrac{3}{4}\overline{\mathbf{e}}_{\mathbf{v}} \rfloor \\
\mathbf{e} &\leftarrow \mathbf{e} + \lfloor \tfrac{1}{2} + \tfrac{3}{8}\overline{\mathbf{f}}_{\mathbf{e}} \rfloor
\end{aligned}$$

This DWT is applied to the mesh that approximates a surface. The resulting vertex coordinates are defined by integer wavelet coefficients that have low absolute values provided the surface is sufficiently smooth. The vertices that

belong to the base mesh represent control points for the coarsest level of detail. Due to the loss of precision in each decomposition step, these are represented by very few bits. The coordinates for all vertices are compressed by arithmetic coding, see [20]. Since many coordinates are zero or have low absolute values, arithmetic coding leads to high compression rates. From a compressed representation we can decode the coordinates and reconstruct an original surface without loss.

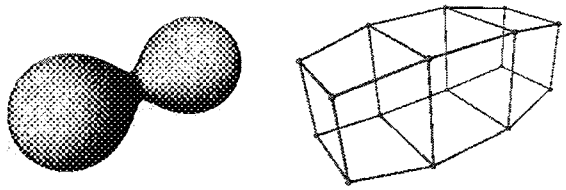

Figure 18: isosurface "two-blobs" and base mesh topology composed of 14 faces and 16 vertices.

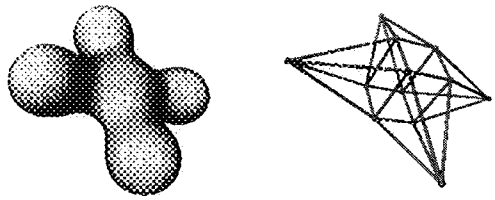

Figure 19: isosurface "five-blobs" and base mesh topology composed of 18 faces and 12 vertices.

We have applied our compression scheme to the isosurfaces "two-blobs" and "five-blobs" obtained from scalar fields, each defined as a sum of gaussians. The two surfaces and their base meshes are depicted in Figures 18 and 19. The isosurfaces are approximated by alternating Catmull-Clark subdivision steps and Newton iteration to project the mesh vertices onto the isosurfaces. The finest level of detail is obtained after five subdivision steps resulting in a total of 14338 vertices for each surface. Every vertex is represented by integer

| wavelet transform | "two-blobs" | "five-blobs" |
|---|---|---|
| linear | 24.6 | 31.0 |
| cubic | 11.8 | 14.8 |
| quintic | 18.8 | 20.5 |
| uncompressed | 172.1 | 172.1 |

Table 4: Storage requirements in kilobytes for surfaces shown in Figures 18 and 19. The uncompressed representation is using four bytes to store a coordinate.

coordinates with a precision of 0.01 percent of the diameter of one gaussian. Lossless compression rates that we obtain with arithmetic coding are listed in Table 4. In addition, one has to store the base mesh connectivity and the histogram of coordinate values for the arithmetic coder.

A different application is the representation of surfaces at multiple levels of detail. Coarse surface representations are defined by wavelet coefficients corresponding to coarse subdivision levels only. Wavelet coefficients on all finer levels are replaced by zero. The reconstruction formula for a DWT defines thus a surface subdivision scheme. Coarse representations for the isosurfaces "two-blobs" and "five-blobs" are shown in Figures 20 and 21. It is not recommended to reduce the precision of scaling function coefficients to display coarse levels of detail. We can either use floating-point arithmetics or integer-arithmetics based on the original lifting parameters from Section 2.

# 5 Conclusions

We have introduced a new wavelet transform defined on arbitrary polygonal meshes, suitable for multiresolution representation and compression of isosurfaces, shock waves and material boundaries. Our method generalizes

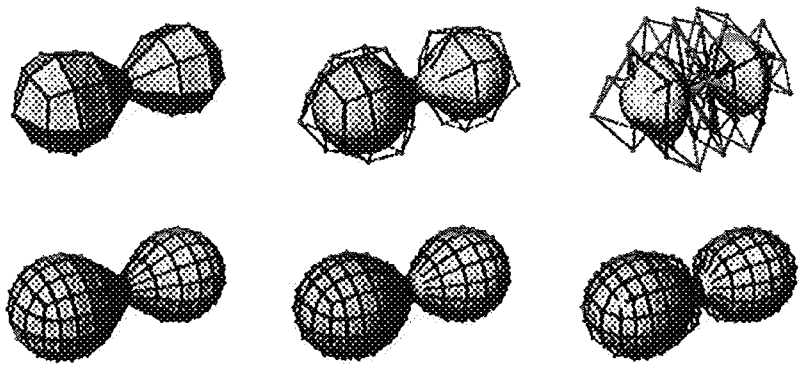

Figure 20: Isosurface "two-blobs" at two different levels of detail (first and second subdivision level) obtained from a floating-point implementation of the DWT. From left to right: linear, cubic, and quintic wavelet transform.

tensor product B-spline surfaces. It is possible to derive a set of B-spline patches from our surface representation at any level of detail. We have also presented a lossless geometry compression scheme based on an integer-to-integer wavelet transform and arithmetic coding. Future work will be directed at efficient and automatic construction of the required base meshes for surfaces defining extremely complex geometry.

# Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract no. W-7405-Eng-48. This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information,

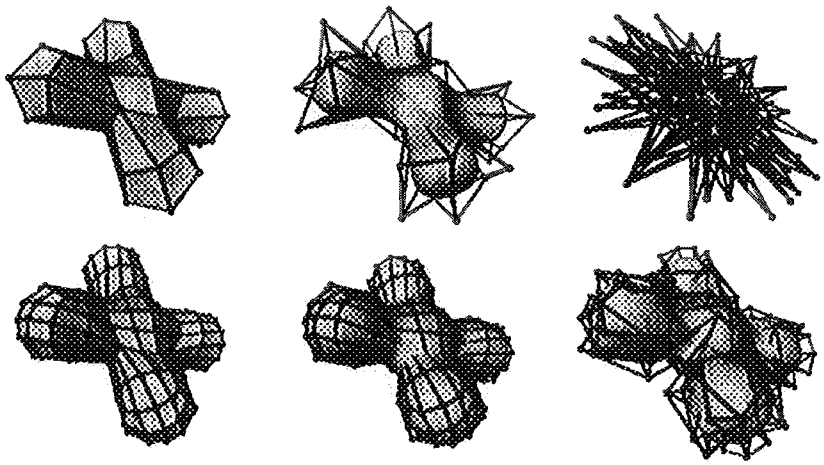

Figure 21: Isosurface "five-blobs" at different levels of detail. Same order of figures as in Figure 20.



This work was supported by the National Science Foundation under contract ACI 9624034 (CAREER Award); the Office of Naval Research under contract N00014-97-1-0222; the Army Research Office under contract ARO 36598-MA-RIP; the NASA Ames Research Center through an NRA award under contract NAG2-1216; the Lawrence Livermore National Laboratory under ASCI ASAP Level-2 Memorandum Agreement B347878 and under Memorandum Agreement B503159; and the North Atlantic Treaty Organization (NATO) under contract CRG.971628 awarded to the University of

California, Davis. We also acknowledge the support of ALSTOM Schilling Robotics, Davis, California; Chevron; and Silicon Graphics, Inc. We thank the members of the Visualization Thrusts at the Center for Image Processing and Integrated Computing (CIPIC) at the University of California at Davis and the Center for Applied Scientific Computing (CASC) at the Lawrence Livermore National Laboratory.

# References

[1] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, *Wavelet transforms that map integers to integers*, Applied and Computational Harmonic Analysis, Vol. 5, No. 3, Academic Press, July 1998, pp. 332–369.

[2] E. Catmull and J. Clark, *Recursively generated B-spline surfaces on arbitrary topological meshes*, Computer-Aided Design, Vol 10, No. 6, Nov. 1978, pp. 350–355.

[3] C.K. Chui, *An Introduction to Wavelets*, Academic Press, San Diego, 1992.

[4] I. Daubechies, *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1992.

[5] I. Daubechies, I. Guskov, P. Schröder, and W. Sweldens *Wavelets on Irregular Point Sets*, Phil. Trans. R. Soc. Lond. A, to appear.

[6] D. Doo and M. Sabin, *Behaviour of recursive division surfaces near extraordinary points*, Computer-Aided Design, Vol 10, No. 6, Nov. 1978, pp. 356–360.

[7] M.A. Duchaineau, *Dyadic Splines*, Ph.D. Thesis, Dept. of Computer Science University of California, Davis, 1996.

[8] N. Dyn, D. Levin, and J.A. Gregory, *A butterfly subdivision scheme for surface interpolation with tension control*, Association for Computing Machinery (ACM) Transactions on Graphics, Vol. 9, No. 2, April 1990, pp. 160–169.

[9] M. Eck and H. Hoppe, *Automatic reconstruction of B-spline surfaces of arbitrary topological type*, Computer Graphics (SIGGRAPH '96 Proc.), Association for Computing Machinery (ACM), 1996, pp. 325–334.

[10] *G. Farin, Curves and Surfaces for CAGD*, Fourth edition, Academic press, San Diego, California, 1997.

[11] B. Guo, *Surface reconstruction: from points to splines*, Computer Aided Design, Vol. 29, No. 4, Elsevier, April 1997, pp. 269–277.

[12] I. Guskov, W. Sweldens, and P. Schröder, *Multiresolution signal processing for meshes*, Computer Graphics (SIGGRAPH '99 Proc.), Association for Computing Machinery (ACM), 1999, pp. 325–334.

[13] B. Jawerth and W. Sweldens, *An overview of wavelet based multiresolution analysis*, Society for Industrial and Applied Mathematics (SIAM) Rev., Vol. 36, No. 3, 1994, pp. 377–412.

[14] L. Kobbelt and P. Schröder, *A multiresolution framework for variational subdivision*, Association for Computing Machinery (ACM) Transactions on Graphics, Vol. 17, No. 4, ACM, Oct. 1998, pp. 209–237.

[15] A.K. Louis, P. Maaß, and A. Rieder, *Wavelets: Theorie und Anwendungen* (in German), B.G. Teubner, Stuttgart, Germany, 1994.

[16] J.M. Lounsbery, *Multiresolution Analysis for Surfaces of Arbitrary Topological Type*, Ph.D. Thesis, Dept. of Mathematics, University of Washington, 1994.

[17] C.T. Loop, *Smooth subdivision surfaces based on triangles*, M.S. Thesis, Dept. of Mathematics, University of Utah, 1987.

[18] S.G. Mallat, *A theory for multiresolution signal decomposition: the wavelet representation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No. 7, 1989, pp. 674–93.

[19] Y. Meyer, *Wavelets, Algorithms & Applications*, Society for Industrial and Applied Mathematics (SIAM), Phiadelphia, 1993.

[20] A. Moffat, R.M. Neal, and I.H. Witten, *Arithmetic coding revisited*, Association for Computing Machinery (ACM) Transactions on Information Systems, Vol. 16, No. 3, July 1998, pp. 256–294.

[21] P. Schröder and W. Sweldens, *Spherical wavelets: efficiently representing functions on the sphere*, Computer Graphics (SIGGRAPH '95 Proc), Association for Computing Machinery (ACM), 1995, pp. 161–172.

[22] T.W. Sederberg, D. Sewell, and M. Sabin, *Non-Uniform Recursive Subdivision Surfaces*, Computer Graphics (SIGGRAPH '98 Proc), Association for Computing Machinery (ACM), 1998, pp. 287–394.

[23] J. Stam, *Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values*, Computer Graphics (SIGGRAPH '98 Proc), Association for Computing Machinery (ACM), 1998, pp. 395–404.

[24] E.J. Stollnitz, T.D. DeRose, and D.H. Salesin, *Wavelets for Computer Graphics, Theory and Applications*, Morgan Kaufmann Publishers Inc., San Francisco, 1996.

[25] W. Sweldens, *Wavelets, Signal Compression and Image Processing*, SIGGRAPH Course Notes, Association for Computing Machinery (ACM), 1994.

[26] W. Sweldens, *The lifting scheme: A custom-design construction of biorthogonal wavelets*, Appl. Comput. Harmon. Anal., Vol. 3, No. 2, pp. 186–200, 1996.

[27] H. Tao and R.J. Moorhead, *Progressive transmission of scientific data using biorthogonal wavelet transform*, in : R.D. Bergeron and A.E. Kaufman, eds., Proceedings IEEE Visualization '94, IEEE Computer Society Press, 1994, pp. 93–99.